

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Previously Presented) A method for delivering keystrokes to descendants of an undecorated inactive window within a Java environment, comprising:
 - remembering a descendant of said undecorated inactive window as a next focus owner when the descendant receives a focus-in event;
 - creating a focus proxy for said undecorated inactive window and issuing a request to set focus to the focus proxy;
 - marking the next focus owner as the true focus owner when the focus proxy receives a focus-in event;
 - directing the focus-in event received by the focus proxy to the true focus owner; and
 - discarding a native focus-out event received by the descendant, wherein the native focus-out event is generated in response to the request to set focus to the focus proxy.
2. (Original) The method of claim 1, wherein the focus-in event received by the descendant is a native focus event generated in response to a request to set focus to the descendant.
3. (Previously Presented) The method of claim 2, wherein remembering the descendant of said undecorated inactive window as the next focus owner comprises discarding the focus-in event received by the descendant.
4. (Previously Presented) The method of claim 2, further comprising discarding a native activation event received by said undecorated inactive window, wherein the native activation event is generated in response to the request to set focus to the descendant.
5. (Currently Amended) The method of claim 1, wherein creating a focus proxy for said undecorated inactive window comprises creating an invisible child of a nearest owning window of the undecorated inactive window that can be active.
6. (Previously Presented) The method of claim 5, further comprising activating the nearest owning window of said undecorated inactive window that can be active.

7. (Original) The method of claim 1, wherein the focus-in event received by the focus proxy is a native focus event generated in response to issuing the request to set focus to the focus proxy.
8. (Canceled)
9. (Previously Presented) The method of claim 1, wherein the descendant of said undecorated inactive window receives the focus-in event in response to a user clicking on the descendant.
10. (Previously Presented) The method of claim 1, wherein the descendant of said undecorated inactive window receives the focus-in event in response to the descendant issuing a focus request through function invocation.
11. (Original) The method of claim 1, wherein the focus proxy receives a key event when a keystroke is delivered to the descendant.
12. (Currently Amended) The method of claim ~~[[11]]~~ 10, further comprising directing the key event to the true focus owner.
13. (Previously Presented) A method for delivering keystrokes to descendants of an undecorated inactive window within a Java environment, comprising:
 - remembering a descendant of said undecorated inactive window as a next focus owner when the descendant receives a focus-in event;
 - creating a focus proxy for said undecorated inactive window and issuing a request to set focus to the focus proxy;
 - marking the next focus owner as the true focus owner when the focus proxy receives a focus-in event;
 - directing the focus-in event received by the focus proxy to the true focus owner; and
 - determining a component gaining focus when the focus proxy receives a focus-out event.
14. (Currently Amended) The method of claim ~~[[13]]~~ 12, further comprising determining whether the component is governed by the focus proxy.
15. (Currently Amended) The method of claim ~~[[14]]~~ 13, further comprising determining whether the component is the true owner if the component is governed by the focus proxy.

16. (Currently Amended) The method of claim ~~[[15]]~~ 14, further comprising discarding the focus-out event if the component is not the true owner.

17. (Currently Amended) The method of claim ~~[[16]]~~ 15, further comprising issuing a request to set focus to the focus proxy.

18. (Currently Amended) The method of claim ~~[[15]]~~ 16, further comprising directing the focus-out event to the true focus owner and setting the true focus owner to null if the component is not governed by the focus proxy.

19. (Currently Amended) The method of claim ~~[[18]]~~ 17, further comprising remembering the component as the next focus owner when the component receives a focus-in event.

20. (Currently Amended) The method of claim ~~[[19]]~~ 18, further comprising issuing a focus request to the focus proxy.

21. (Currently Amended) The method of claim ~~[[20]]~~ 19, further comprising marking the next focus owner as the true focus owner when the focus proxy receives a focus-in event.

22. (Currently Amended) The method of claim ~~[[21]]~~ 20, further comprising directing the focus-in event delivered to the focus proxy to the true focus owner.

23. (Previously Presented) A method for delivering keystrokes to descendants of an undecorated inactive window within a Java environment, comprising:

remembering a descendant of said undecorated inactive window as the next focus owner

when the descendant receives a focus-in event;

creating a focus proxy for said undecorated inactive window and issuing a request to set

focus to the focus proxy;

marking the next focus owner as the true focus owner when the focus proxy receives a

focus-in event;

directing the focus-in event received by the focus proxy to the true focus owner;

directing a key event received by the focus proxy to the true focus owner; and

discarding a native focus-out event received by the descendant, wherein the native focus-out event is generated in response to the request to set focus to the focus proxy.

24. (Currently Amended) The method of claim ~~[[23]]~~ 22, wherein the key event is generated in response to a user delivering keystrokes to the descendant.
25. (Previously Presented) A method for delivering keystrokes to descendants of an undecorated inactive window within a Java environment, comprising:
 - creating a focus proxy for said undecorated inactive window and directing a key event received by the focus proxy to a descendant of said undecorated inactive window;
 - and
 - discarding a native focus-out event received by the descendant, wherein the native focus-out event is generated in response to the request to set focus to the focus proxy.
26. (Currently Amended) The method of claim ~~[[25]]~~ 24, wherein the focus proxy is created when the descendant receives a focus-in event.
27. (Currently Amended) The method of claim ~~[[26]]~~ 25, wherein the descendant receives a focus-in event in response to a user clicking on the descendant.
28. (Currently Amended) The method of claim ~~[[26]]~~ 25, wherein the descendant receives a focus-in event in response to the descendant issuing a focus request through function invocation.
29. (Original) The method of claim 25, wherein the focus proxy receives the key event when a keystroke is delivered to the descendant.
30. (Currently Amended) The method of claim ~~[[29]]~~ 28, wherein creating the focus proxy comprises determining the nearest owning window of said undecorated inactive window that can be active and creating an invisible child of the nearest owning window of said undecorated inactive window that can be active as the focus proxy.
31. (Previously Presented) A computer-readable medium having stored thereon a program which is executable by a processor, the program comprising instructions for:
 - creating a focus proxy for an undecorated inactive window within a Java environment and directing a key event received by the focus proxy to a selected descendant of said undecorated inactive window; and
 - discarding a native focus-out event received by the descendant, wherein the native focus-out event is generated in response to the request to set focus to the focus proxy.

32. (Currently Amended) The computer-readable medium of claim ~~[[31]]~~ 30, wherein the program creates the focus proxy in response to the descendant receiving a focus-in event.
33. (Currently Amended) The computer-readable medium of claim ~~[[32]]~~ 31, wherein the descendant receives the focus-in event in response to a user clicking on the descendant.
34. (Currently Amended) The computer-readable medium of claim ~~[[32]]~~ 31, wherein the descendant receives the focus-in event in response to the descendant issuing a focus request through function invocation.
35. (Original) The computer-readable medium of claim 31, wherein the program further comprises instructions for remembering the descendant as the next focus owner when the descendant receives the focus-in event.
36. (Currently Amended) The computer-readable medium of claim ~~[[35]]~~ 31, wherein the program further comprises instructions for marking the next focus owner as the true focus owner when the focus proxy receives a focus-in event.
37. (Currently Amended) The computer-readable medium of claim ~~[[36]]~~ 31, wherein the program further comprises instructions for directing the focus-in event received by the focus proxy to the true focus owner.
38. (Previously Presented) The computer-readable medium of claim 31, wherein instructions for creating a focus proxy for said undecorated inactive window comprise instructions for determining the nearest owning window of said undecorated inactive window that can be active and creating an invisible child of the nearest owning window that can be active as the focus proxy.